**IT-Колледж “Сириус”**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ДОКЛАД**

по дисциплине “Web”

на тему “История развития Web технологий”

Выполнил:

Студент группы

1.9.7.1

Василенко Дмитрий Евгеньевич

Принял:

Старший преподаватель

Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”

2022

**Язык гипертекстовой разметки HTML и первые сайты**

Язык гипертекстовой разметки HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1991 году в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. Свои первые черты язык начал обретать в 1986 году. Толчком стало принятие Международной организацией по стандартизации (ISO) ISO-8879-стандарта - Standard Generalized Markup Language или, в сокращенном варианте - SGML. К нему прилагалось описание, в котором говорилось о том, что SGML предназначен для структурной разметки текста. Примечательно, что описания внешнего вида документа не предполагалось.



Первый в мире [сайт](http://info.cern.ch/) появился 6 августа 1991 года. Его создатель, Тим Бернерс-Ли, опубликовал на нём описание новой технологии World Wide Web, основанной на протоколе передачи данных HTTP, системе адресации URI и языке гипертекстовой разметки HTML. Также на сайте были описаны принципы установки и работы серверов и браузеров. Сайт стал и первым в мире интернет-каталогом, так как позже Тим Бернерс-Ли разместил на нём список ссылок на другие сайты. Последующие сайты представляли собой набор текстовых страниц, связанных встроенными ссылками вплоть до 1992 года. Впоследствии в 1993-1994 годах были разработаны сайты на основе таблиц, которые предлагали лучшее размещение контента и усовершенствованный стиль навигации.

**Язык таблицы стилей CSS**

Около сорока тегов содержал HTML версии 1.2. Описания физического представления документов также не имелось. Как и его прародитель - SGML, он был преимущественно ориентирован на логическую и структурную разметку текста. Впрочем, некий намек на то, как будет физически представлена страница, ряд тегов все же делал.

Разработкой HTML версии 2.0 занялся консорциум W3С. Первый результат удалось получить, спустя год насыщенной работы - в 1995 году. Практически параллельно обсуждались возможности версии 3.0. Если вторую версию нельзя назвать существенно отличающейся от первой, то третья стала безусловным прорывом.

HTML 3.0 включал интересные новинки:

* Разметку математических формул
* Теги для создания страниц
* Вставку рисунков, обтекаемых текстом
* Примечаний и т.д.

Однако, этого было недостаточно, потребность в визуальном оформлении гипертекстовых страниц становилась все более актуальной. Тогда, W3С приступили к созданию самостоятельной системы, при этом не противоречащей основам HTML, но позволяющей описывать визуальное оформление документов. Результатом стало появление в 1996 году CSS - Cascading Style Sheets, иерархические стилевые спецификации, наделенные уникальным синтаксисом, структурой, задачами.

Почему же сегодня все чаще используется CSS? Потому что HTML, не смотря на свои существенно расширившиеся с момента создания возможности, остается языком логической разметки гипертекста, т.е. не связан с оформлением документа. Современные стандарты интернета подразумевают создание ярких и запоминающихся страниц, поэтому веб-мастера все чаще используют CSS. Можно ли ставить точку в истории HTML? Ответ на этот вопрос будет, скорее положительным, но полностью язык не исчезнет, т.к. он лежит в основе многих других систем.

Стоит также упомянуть, что на данный момент большинство использует новейшую открытую платформу HTML5 в связке с каскадными стилями 3-го поколения CSS3.



**PHP & JavaScript**

**PHP**

На сегодняшний день, язык программирования PHP является мощнейшим инструментом для разработки Веб-приложений. Простота языка, богатые возможности, широкая функциональность делает его самым популярным языком у Веб-разработчиков. Кроме того PHP был разработан в рамках проекта Open Source. Поэтому интерпретатор PHP есть практически на каждом хостинге. Язык PHP очень простой в изучении, на просторах интернета выложены различные полезные библиотеки на данном языке, присутствует множество примеров реализации различных функций. Начинающему PHP программисту достаточно «загуглить» строчку «как … сделать на PHP», и наверняка он сразу найдет подробный пример или универсальную библиотеку для реализации требуемого действия.

**История PHP** начинается в 1994 году, когда программист энтузиаст Расмус Лердорф сделал набор из скриптов на языке Perl. Цель этой разработки — сохранение статистики просмотров его резюме в формате html. Расмус наименовал свое творение Personal Home Page (Персональная Домашняя Страница). Когда функциональности и быстроты языка Perl для его проектов стало не хватать, Лердорф написал новый интерпретатор на языке C. Так появилась первая версия интерпретатора PHP.

Спустя 3 года была разработана вторая версия интерпретатора PHP, которая также была написана на языке C. Популярность PHP в то время была крайне низкой — около 5 десятков тысяч человек по всему миру использовали его для своих разработок.

Третья версия **PHP вошла в историю**, именно она определила современный стиль и облик PHP. Переработанная израильскими программистами Энди Гутмансом и Зеевом Сураски версия PHP 3.0 официально вышла летом 1998 года.

Основное преимущество PHP 3.0 — возможность подключения к ядру дополнительных модулей для расширения функционала. Язык PHP набирал все больше сторонников. Он поддерживал работу со всеми популярными базами данных. В PHP 3.0 была возможна поддержка множества протоколов и различных API. Рост популярности языка PHP способствовал его стремительному развитию. PHP стал расшифровываться как «hypertext Preprocessor» — гипертекстовый препроцессор.

В конце 1998 года израильские программисты начали разрабатывать новый движок. Они сделали акцент на увеличение производительности и совершенствование модульности базиса PHP кода. Благодаря труду разработчиков в 1999 году родился движок под названием Zend Engine, который был на голову выше движка предыдущей версии PHP. На основе Zend Engine к 2000 году официально был выпущен язык PHP 4.0. В новой версии значительно возросла производительность, присутствовал ряд полезных нововведений. В PHP 4.0 была включена поддержка сессий. Стало возможно буферизировать вывод. Язык стал более безопасный.

В середине июля 2004 года вышла пятая версия PHP. PHP 5 работает на ядре Zend Engine 2, которое значительно эффективней и производительней. Значительным моментом в *истории развития PHP* было внедрения полноценного объектно-ориентированного программирования в PHP 5, что в разы упростило и сделало удобным разработку масштабных проектов на языке PHP. В настоящее время последняя версия интерпретатора PHP — версия 5.4.

Шестая версия находится в разработке с 2006 года. Но спустя 4 года разработчики посчитали ее бесперспективной из-за возникших проблем. Судьба PHP 6 пока что неизвестна.РР

**JavaScript**

JavaScript, вероятно, является одним из самых важных языков программирования на сегодняшний день. Благодаря развитию веба JavaScript достиг высот, которых ему не сулили.

События, в результате которых появился JavaScript, разворачивались в течение шести месяцев, с мая по декабрь 1995 года. Компания Netscape Communications уверенно прокладывала себе путь в области веб-технологий. Её браузер Netscape Communicator успешно отвоевывал позиции у NCSA Mosaic, первого популярного веб-браузера. Netscape была создана людьми, принимавшими участие в разработке Mosaic в ранние 90-е. Теперь, с деньгами и независимостью, у них было всё необходимое для поиска способов дальнейшего развития веб-технологий. Именно это послужило толчком для рождения JavaScript.

Отец JavaScript Брендан Айк должен был разработать для Netscape «Scheme для браузера». Scheme — это динамичный, мощный и функциональный диалект языка программирования Lisp с максимально упрощённым синтаксисом. Вебу требовалось что-то подобное: простое в освоении, динамичное, немногословное и мощное. Айк не стал упускать шанса поработать над тем, что ему нравилось, и присоединился к команде.

Перед командой была поставлена задача подготовить работающий прототип в кратчайшие сроки. Sun Microsystems заканчивала работу над своим языком программирования Java, на тот момент называвшимся Oak, и Netscape Communications была уже готова заключить с компанией контракт, чтобы сделать Java доступным в своем браузере. Так зачем же понадобился Mocha (первое название JavaScript)? Зачем нужно было создавать абсолютно новый язык программирования при наличии готовой альтернативы? Дело в том, что Java не был предназначен для той аудитории, на которую ориентировался Mocha — скриптеры, любители, дизайнеры. Java был слишком большим и навороченным для того, чтобы выполнять эту роль. Основная идея заключалась в том, что Java должен был предназначаться для крупных разработчиков и профессиональных программистов, в то время, как Mocha должен был использоваться для небольших скриптовых задач. Другими словами, Mocha должен был стать скриптовым компаньоном для Java по принципу, аналогичному тому, как взаимодействуют C/C++ и Visual Basic на платформе Windows.

*Инженеры Netscape приступили к детальному изучению Java. Они даже начали разрабатывать собственную виртуальную машину Java, однако проект быстро свернули, так как она не могла достичь идеальной совместимости с виртуальной машиной Sun Microsystems.*  
  
 *Проблема скорейшего выбора языка стояла как никогда остро. Возможными кандидатами были Python, Tcl и Scheme. Айк должен был действовать быстро. По сравнению с конкурентами у него были два преимущества: свобода в определении набора необходимых возможностей и прямая связь с заказчиком. К несчастью, имело место и очевидное неудобство: для принятия огромного количества важных решений времени практически не было. JavaScript, a.k.a. Mocha, был рождён именно в таких условиях. В течение нескольких недель был подготовлен рабочий прототип, который затем был интегрирован в Netscape Communicator.*  
  
 *То, что должно было стать аналогом Scheme для браузера, вылилось в нечто совершенно иное. Рукой Айка управляли необходимость закрыть сделку с Sun и сделать Mocha скриптовым компаньоном для Java. Синтаксис должен был быть максимально близким Java. Помимо этого, от Java была унаследована семантика для большого количества устоявшихся идиом. Таким образом, Mocha был совсем не похож на Scheme. Он выглядел, как динамический Java, под оболочкой которого скрывался гибрид Scheme и Self.*  
  
 *Прототип Mocha был интегрирован в Netscape Communicator в мае 1995 года. Через очень короткий промежуток времени он был переименован в LiveScript, так как в тот момент слово live выглядело очень привлекательным с точки зрения маркетологов. В декабре 1995 года сделка между Netscape Communications и Sun была закрыта: Mocha/LiveScript был переименован в JavaScript и преподносился в качестве скриптового языка для выполнения небольших клиентских задач в браузере, в то время, как Java был полноценным профессиональным языком программирования для разработки сложных веб-компонентов.*  
  
 *Первая версия JavaScript заложила все те фундаментальные особенности, которыми этот язык знаменит и поныне. В частности, его объектная модель и функциональные особенности уже присутствовали в первой версии.*

Различные реализации

Когда Sun и Netscape закрыли сделку, и Mocha/LiveScript был переименован в JavaScript, встал ребром очень важный вопрос: что будет с конкурентами? Хоть Netscape и набирал популярность, становясь самым используемым браузером, Microsoft занималась активной разработкой Internet Explorer. С самых первых дней JavaScript показал настолько удивительные возможности в плане взаимодействия с пользователем, что соперничающим браузерам не оставалось ничего иного, кроме как в кратчайшие сроки найти готовые решения, представлявшие собой рабочие реализации JavaScript. В тот момент (и ещё достаточно долго после этого) веб-стандарты оставались достаточно слабыми. Поэтому Microsoft разработала свою реализацию JavaScript, назвав ее JScript.  
  
 Реализация JavaScript получила свое собственное название и в Netscape. Версия, выпущенная вместе с Netscape Navigator 2.0, была известна, как Mocha. Осенью 1996 года Айк переписал бóльшую часть Mocha, чтобы разобраться с техническими огрехами и недоработками, возникшими, как следствие спешки при разработке. Новая версия была названа SpiderMonkey. Это название используется по сей день в JavaScript-движке браузера Firefox, внука Netscape Navigator.  
  
  
  
ECMAScript: стандартизированный JavaScript

Первой большой переменой для JavaScript после его выпуска стала стандартизация ECMA. ECMA — ассоциация, созданная в 1961 году с целью стандартизации информационных и коммуникационных систем.  
  
 Работа над стандартизацией JavaScript началась в ноябре 1996 года. Стандарту, над которым работала группа TC-39, был присвоен идентификационный номер ECMA-262. К тому моменту JavaScript активно использовался на многих веб-страницах. В [этом пресс-релизе](https://web.archive.org/web/19981203070212/http://cgi.netscape.com/newsref/pr/newsrelease289.html) 1996 года указано количество в 300000 страниц, использующих JavaScript.

JavaScript и Java лежат в основе Netscape ONE — платформы для разработки интернет- и интранет-приложений. В течение короткого времени с момента своего внедрения в прошлом году новые языки получили признание среди разработчиков, благодаря чему на данный момент существует более 175000 Java-апплетов и более 300000 страниц, использующих JavaScript по всему интернету, согласно [www.hotbot.com](http://www.hotbot.com/). — [Netscape Press Release](https://web.archive.org/web/19981203070212/http://cgi.netscape.com/newsref/pr/newsrelease289.html)

Стандартизация стала для молодого языка не только важным шагом, но и серьезным вызовом. Она открыла JavaScript для большей аудитории и дала возможность сторонним разработчикам принимать участие в развитии языка. Она также помогла держать других разработчиков в рамках. В те времена бытовало опасение, что Microsoft или кто-либо ещё могут слишком сильно отклониться от оригинальной реализации языка, что могло привести к фрагментации.  
  
 Из-за проблем с торговой маркой ECMA не могла использовать JavaScript в качестве названия. После непродолжительных дебатов было решено, что описанный стандартом язык программирования будет назван ECMAScript. На сегодняшний день JavaScript это всего лишь коммерческое название ECMAScript.

**Рождение AJAX**

AJAX (asynchronous JavaScript and XML) — технология, появившаяся на свет в годы ECMAScript 3. Хоть она и не являлась частью стандарта, Microsoft встроила некоторые расширения для JavaScript в Internet Explorer 5. Одним из таких расширений была функция XMLHttpRequest в виде управляющего элемента ActiveX XMLHTTP. Эта функция позволяла браузеру выполнять асинхронные HTTP-запросы серверу, тем самым позволяя страницам обновляться на лету. Хотя само название AJAX было придумано значительно позже, сама техника активно использовалась в то время.

Впервые термин AJAX был использован Джесси Джеймсом Гарреттом в его [статье](http://adaptivepath.org/ideas/ajax-new-approach-web-applications/).

Заключение

История JavaScript длинна и полна неожиданных поворотов. Благодаря Node.js, V8, Jquery, Express, Next, Vue и другим проектам JavaScript поднялся на высоты, о которых разработчики первой версии даже не задумывались. Активные предложения, пребывающие в разных стадиях, делают будущее JavaScript чистым и безоблачным. Пройдя долгий путь, полный неожиданных поворотов и препятствий, JavaScript остаётся одним из самых успешных языков в истории программирования. И это — лучшее доказательство его надежности. Всегда ставьте на JavaScript.



(Скриншот сделан на сайте [octovers](https://octoverse.github.com/))

**Развитие web-серверов**

Изначально, основой Всемирной паутины были web-сервера CERN httpd, написанные Тимом Бернерсом-Ли на языке программирования Си. Сервер NCSA httpd появился после CERN, когда возникла потребность в небольшом и быстром web-сервере. Позднее к проекту подключились другие авторы, которые стали накладывать свои патчи (программы-дополнения). В 1995 году Брайан Белендорф объединил эти патчи и создал первую версию сервера **Apache** (сокращение от «a patchy server»), который по настоящее время занимает лидирующую позицию по популярности.

Первая версия имела номер 0.6.2. В декабре 1995 года вышел релиз Apache 1.0. В следующие годы группа разработчиков расширилась, и Apache был портирован на различные операционные системы.

В 1999 году была создана некоммерческая организация Apache Software Foundation, а в марте 2000 года состоялась первая конференция разработчиков под названием ApacheCon, где была представлена версия Apache 2.0. В ней была переработана предыдущая серверная архитектура.

На сегодняшний день существуют 2 ветки Apache — 1.3 и 2.0.  
Одновременно с появлением Apache, компания Microsoft выпустила коммерческий web-сервер Internet Information Server (IIS), как надстройку для операционной системы Windows NT 3.51. В этой версии отсутствовали многие возможности, но имелись стандартные веб-службы, FTP и Gopher. В более поздней версии IIS3, вышедшей в 1997 году, появились новые возможности, такие как Active Server Pages (ASP), а через год в IIS4 упразднена служба Gopher и значительно расширен набор параметров.  
В 2003 году вышел web-сервер IIS6 в составе операционной системы Windows Server 2003. В данной версии произошли значительные изменения: добавлена поддержка программной платформы .NET и применен новый подход с точки зрения безопасности.

С момента появления по настоящее время IIS является вторым по популярности web-сервером. Остальные существующие сервера созданы для выполнения узкопрофильных задач и пользуются сравнительно небольшой популярностью. Например, программа HTTP File Server, написанная в 2002 году Массимо Мелиной, работает, как простой сервер для передачи файлов по сети, а Lighttpd (2003) создана для использования на сильно нагруженных системах.

О других Web-технологиях

**TypeScript**

TypeScript — это открытый язык программирования, разработанный Microsoft. Его итоговый код перед выполнением компилируется все в тот же JavaScript. Начиная с его появления в 2012 году, этот язык продолжает активно развиваться, и с каждым годом его популярность в мировом сообществе растет.

На ранних этапах этот язык сильно ассоциировался с Angular, поскольку по умолчанию Angular предполагает использование именно TS, хотя это можно перенастроить.

Синтаксис TypeScript позволяет разработчику элегантно выразить решение задачи в виде текста в файле . ts или . tsx . Он является эволюцией синтаксиса JavaScript, поэтому любая программа JS синтаксически корректна на TypeScript. Компилятор tsc помогает обнаружить множество дефектов до загрузки на продакшн. Он преобразует исходный текст TypeScript в JavaScript и анализирует программу, стараясь найти проблемные места. TypeScript может создать. js-файлы для любой версии JavaScript, начиная с ES3. Разработчик может диктовать TypeScript, какие правила анализа активировать, а какие — отключить.

**SASS**

Sass — это метаязык (язык для описания другого языка), который упрощает и ускоряет написание CSS-кода. Его часто называют препроцессором CSS — это означает, что Sass имеет свой синтаксис, из которого генерирует CSS-код, понятный любому браузеру.

Препроцессор Sass помогает:

* сделать CSS-код понятнее и проще. Его легче масштабировать, обновлять и поддерживать;
* расширить функциональность. С помощью Sass можно использовать CSS-константы, встроенные функции, вложенные правила, примеси (смешанные стили), наследование и так далее;
* избежать многократного повторения одинаковых фрагментов кода. Это экономит время разработчика, уменьшает объем файлов стилей и ускоряет обработку страниц.

SCSS

SCSS — "диалект" языка SASS. Отличие SCSS от SASS заключается в том, что SCSS больше похож на обычный CSS код

Css Фреймворки

Для того, чтобы избежать этого большого количества работы и ускорить процесс верстки веб-страниц, были созданы специальные библиотеки или CSS-фреймворки. По сути, CSS-фреймворк - это просто файл CSS, который вы стандартным образом подключаете к вашей веб-странице.

Этот CSS файл просто содержит уже написанный за вас набор стилей, которые вы сможете применять к элементам на вашем веб-сайте, после того, как эта библиотека будет подключена к вашей веб-странице.

После того, как этот файл фреймворка будет подключен к веб-странице, вы можете добавлять к вашим элементам ни какие-то конкретные стили CSS, а просто добавлять классы.

Наиболее популярными css фреймворками являются:

* Bootstrap
* Foundation
* Bulma
* Skeleton

Вывод:

История web-технологий наполнена множеством странных, и неожиданных поворотов. В этом докладе я назвал лишь несколько технологий, о которых мне известно и безусловно, сфера Web - это огромный стек технологий, для разработки множества сервисов различной сложности, в которой JS, HTML, CSS и все вышеперечисленные технологии- лишь вершина айсберга